Delta catheti (hard)

(3, 4, 5) is a famous Pythagorean triple, it gives a quick answer to the question:
For a given integer **d**, is there a Pythagorean triple **(a, b, c)** such that **b - a = d**?

A solution is **(3d, 4d, 5d)**, and in fact one can easily prove that the set of solutions is infinite, and that there is an obvious total order on those solutions.
Given **n**, you'll have to find the **n**th term of the sequence of solutions.

Geometrically, it is the study of right triangles for which the difference of the catheti are equal to **d**.

Input

The first line of input contains an integer **T**, the number of test cases.

Each of the next **T** lines contains three integers **n, d, m**.

Output

For each test case, compute the **n**th term amongst the solutions **(a, b, c)** for the problem :
$a^2 + b^2 = c^2$ with **b - a = d** and **0 < a < b < c** .

As the answer could not fit in a 64-bit container, simply output your answer modulo **m**.

Example

Input:
```
3
1 1 235813
3 21 1000
9 119 11
```

Output:
```
3 4 5
63 84 105
5 3 1
```

Explanations

For the first case, the first solution is **(3, 4, 5)**, as 4 - 3 = 1.

For the second case, the firsts solutions are:
(15, 36, 39), (24, 45, 51), (63, 84, 105), (144, 165, 219), (195, 216, 291), (420, 441, 609), ...
The third one is **(63, 84, 105)**.

For the third case, the first solutions are:
(24, 143, 145), (49, 168, 175), (57, 176, 185), (85, 204, 221), (136, 255, 289), (180, 299, 349), (196, 315, 371), (261, 380, 461), (357, 476, 595), (481, 600, 769), (616, 735, 959), ...
The 9th solution is **(357, 476, 595)**, reduced modulo 11, we get **(5, 3, 1)**.

Constraints

0 < T < 10^5
0 < n < 10^7
0 < d < 10^8
1 < m < 10^9

n, d, m : Uniform randomly chosen in their range.

Constraints allow some interpreted languages to get AC, but it could be hard.
For your information, I have two distinct solutions.
My first python3 code get AC under 32s with 28MB of memory print.
My second python3 code get AC under 16s with 16MB of memory print.
With a compiled language, it should be at least 20× faster. Have fun ;-)